



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/575,342	05/19/2000	Jouni Rapakko	460-009420-US(PAR)	9989
7590	07/12/2006		EXAMINER	
Clarence A Green Perman & Green LLP 425 Post Road Fairfield, CT 06430			VU, TUAN A	
			ART UNIT	PAPER NUMBER
			2193	

DATE MAILED: 07/12/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/575,342	RAPAKKO ET AL.	
	Examiner Tuan A. Vu	Art Unit 2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 07 June 2006.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-28 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-28 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) All b) Some * c) None of:
1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____ | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 6/07/2006.

As indicated in Applicant's response, claims 1, 7, 14-15, 21, and 28 have been amended.

Claims 1-28 are pending in the office action.

Specification: Abstract

2. The abstract of the disclosure is objected to because it contains numerals within parentheses and a bottom reference to a Figure, which is rather more appropriate for a PCT than a standard US application. Cleaning of these informalities is required. See MPEP § 608.01(b).

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claim 1-28 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 1 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite or unclear in laying out essential structural cooperative relationships of elements, such lack of clarity amounting to a hard to follow relationship between the necessary structural connections. See MPEP § 2172.01.

The claim recites 'loading, starting, and executing program modules ... the user interface software is divided ... basic module ... user interface module ... parts of the same interface software ... and being stored within the electronic device; ... executing the loading of the ... software in at least two phases ...'; the confusing relationships can be described in two parts:

(i) when it is construed that the user interface software has two parts, the basic module and the user interface module, the reciting of ‘and being stored within the electronic device’ does not make it clear what actually has been currently stored in the electronic device system when the step of loading and starting occurs (i.e. *loading, starting and executing program modules...*). That is, this ‘being stored’ leads to more than one interpretations; e.g. EITHER both parts of the user interface (UI) software are already resident but unrelated to this loading/starting; OR that they are currently resident in the system for the loading/starting to actualize; OR that the loading would have for result that these parts (UI module and basic module) be stored in the device.

(ii) the recited step as to ‘executing the loading of the user interface software in at least two phases’ makes it unclear as whether or not the loading/starting as mentioned in (i) is the same process by which the UI software is loaded from the expansion card, or the process at the start of which the UI software modules (both modules or just one?) are already stored in the resident system of the target device. That is, ‘the loading’ here refers to – or lends to the interpretation that it does so-- the very loading (i.e. *loading, starting and executing program modules...*) from above, thus it is reasonable to construe that it is the effect of the 2-phase (*executing the loading ...*) operations that enables the storing of the parts as mentioned in (i), making the ‘being stored’ limitation from above unclear as to exactly what is being stored, and if so, before or after the card insertion phase occurs. If only the basic module in the first phase is considered being stored by the time the second phase happens, then the recited ‘the loading’ in (ii) does not make it clear as to what loading actually consists of, and where the basic module (or the UI module) is physically located when this loading executes -- when *the loading* in (ii) is perceived as conveying that both parts of the UI software are currently in the device system via

the interpretation that it is the loading in (i). From the claim language describing the second phase, it can be interpreted that the loading of the basic module would exclude the UI module being resident in the system, which is not what the limitation ‘and being stored within the electronic device’ from above entails. Further, the claim recites that the UI software in the expansion card is divided in distinct parts, the UI module and the basic module; and if the basic module is construed as being a part of this software in the card, then the second phase would be hard to construe because loading of the 1st phase (the basic module) would be interpreted such that this first phase needs a card insertion as well, which again contradicts with the ‘being stored’ as in (i) when the 2nd phase does not strongly entail that the card is to be inserted only as a first time.

The claim entails that the basic module is part of the UI software in the expansion card, but make it unclear as to the physical whereabouts of this module when the loading starts; and this also lends to more than one interpretations: relative relationship between the modules and the execution engine, the memory storage and the expansion card seem to generate more confusion as to their interrelated location and timing. For example, the claimed loading step as in (i) -- which is that of (ii) – can be interpreted as a loading sequence in which a basic module is loaded from unclear origins into the device to detect the insertion of the second phase and the loading of UI module, both being stored in the device. Further, the limitation as to the 2 parts of the UI software requires clarification as to whether the parts are integral to the UI software as this software is loaded as from an expansion card (*method for loading the user interface software of an expansion card in a ... device*); or that they are physically divided/separated; and if so, how the implementation of this separation would be claimed so to convey some reasonable

sequence of actions therein, e.g. that at the time of the loading one module already exists as resident in the system.

The lack of clarity in the loading as set forth in the observations above will be addressed with the following interpretation. The basic module being loaded does not require that it come from a prior loading from any particular definite source medium to make it readily running when the second phase starts; and that the loading and execution of the UI software -- as recited in (i) and (ii) – amounts to a loading process with 2 phases having as one end result that both UI module and basic module ‘are being stored’ in the system.

Claims 2-6 are rejected for not remedying to the deficiency of the base claim.

Claim 7, 14-15, 21 and 28 also recite ‘and being stored within the electronic device’ in conjunction with a subsequent recital of ‘the loading of the … interface module’ (claims 7, 21), or ‘executing the loading…’ (claim 15), or ‘the loading … comprises’ (claims 14, 28); thus are rejected for the same reasons as set forth above.

Dependent claims 8-13, 16-20, 22-27 are also rejected for failing to remedying that deficiency.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. Claims 1-28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shih et al., USPN: 6,405,362 (hereinafter Shih), in view of Garney, USPN: 5,319,751 (hereinafter Garney).

As per claim 1, Shih discloses a method for loading the software application from an expansion card in an electronic device (e.g. col. 3, lines 8-18), said method comprising loading, starting and executing program modules in the device (Fig. 2,3); which expansion card can be coupled in a releasable manner to the device (Fig. 1; 3); executing the loading of the expansion card application is done in 2 phases (Fig. 2, 3); wherein the first phase includes the loading and start-up of the basic module (e.g. *event monitor* , step 300 - Fig. 2; col. 6, lines 25-37); and

the second phase includes conducting the loading and start-up of the software application module (e.g. steps 315-320 – Fig. 3) when the expansion card is coupled to the electronic device, the basic module already loaded (Note: event monitor to see if insertion or removal event occurs reads on its being loaded prior to such event);

the basic module receiving a signal about attaching the expansion card and loads the software application (Fig. 3); and

the basic module and the card application being stored within the electronic device (Note: basic module and card application being loaded are construed as being stored in the loading process).

Shih discloses software applications on hand held devices or user event-driven applications, each of them necessarily include user interface modules (e.g. col. 6, lines 5-32); hence has disclosed that the expansion software application to be loaded can be an user interface software or application modules used in such handheld devices.

Nor does Shih specify that said user interface software from the expansion card is divided in a basic module and a user interface module, said basic module and said user interface module

being separate parts of the same user interface software. The loading and activation of software from removable media being divided into a basic operating system level module and a main software module was a known concept in the art of computer booting and loading of operating system components or upgradable software. And this has been applied to software provided via removable device being coupled to host computer system targeted for the purpose to load up thereto such additional software or O.S. components. Whereas Shih uses a shell stand-alone code or thread to snoop on card insertion event, the importance of straining on the operating system resources or obviating tight dependency on architecture diversity and crash due to conflict thereof is evidenced (see Shih: col. 6, line 41 to col. 7, line 18; col. 30-46; col. 9, line 24-53) via provision by the card insertion with module or OS related code to support the specifics of the software to install with such diversity and compatibility of resources. Analogous to the endeavor so concerned by Shih, Garney, in a method to activate/configure a processing device with software loaded from the removable resources being attached thereon analogous to the expansion card loading by Shih, teaches the software loading being done in 2 parts, with both parts being separate components of the software product; the first part being a stub for setting basic operating system configuration for preparing the host machine to incorporate the second part on the software which is loading and activation the content of the software in the removable card (Fig. 6-12). At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to provide a software product for such product loading method as suggested by Shih with a 2 separate parts or components just as taught by Garney just so to prepare the host target system with first software basic (Garney's stub) component to incorporate a second and main component of the software for which more resources are required

and yet assure secure control and operation of the incorporation of this main software component in that less resources would be used to forestall conflicts by which more resources could be otherwise jeopardized.

As per claim 2, Shih further teaches a method wherein the first basic module controls the execution of the second phase (step 315 – Fig. 3).

As per claim 3, Shih further teaches application programming interface and device driver to arrange communication between user interface software and expansion card (Fig. 3 – Note: O.S. device driver recognition cooperating with shell or windows API for signaling an hardware insertion is implicitly disclosed); said basic module being signaled on the coupling of the expansion card from device driver for effecting the loading and start-up of the software user interface module (e.g. col. 6, line 20 to col. 8, line 32).

As per claim 4, Shih does disclose wherein coupling an expansion card to a electronic device an interrupt signal is produced with OS examination of cause therefor; and information on the coupling is transmitted to a device driver (e.g. col 7, lines 30-50 – Note: Shell notifying a event monitor to allocate immediate resource for handling a signal is equivalent to interrupt signal handler for addressing hot insertion of card).

As per claim 5, Shih discloses wherein the decoupling of an expansion card halts processing of a user interface module without interrupting the basic module (Fig. 2,3 – Note: event monitor is required to remain functional even if card is removed for signal removal event).

As per claim 6, Shih discloses that memory is allocated for a user interface module when said module is loaded and said memory is deallocated when an expansion card removed from an electronic device (e.g. col. 7, lines 19-23, 62-67)

As per claim 7, Shih discloses a electronic device comprising means for loading a expansion card software in an electronic device, means for coupling an expansion card in a releasable manner in electronic device; and means for loading, starting and executing program modules in the device (Fig. 2,3); and the loading of the application being arranged to be executed when the expansion card is coupled to the device and the basic module already loaded; (Note: event monitor to see if insertion or removal event occurs reads on its being loaded prior to such event); wherein a basic module receives a signal about attaching the expansion card and loads the software application (Fig. 3); the basic module and the card application being stored within the electronic device (Note: basic module and card application being loaded are construed as being stored in the loading process).

From claim 1, Shih discloses that the expansion software application to be loaded can be an user interface software or application modules (e.g. col. 6, lines 5-32) used in such handheld devices.

Nor does Shih disclose that the user interface software is divided in a basic module and a user interface module, said basic module and said user interface module being separate parts of the same user interface software; but this limitation has been addressed in claim 1 above using Garney.

As per claims 8-9, these are the apparatus claims corresponding to the method of claims 2-3, respectively. The claims are rejected under the same arguments as cited above, with Column 2, Line 1 referencing the apparatus (information process apparatus).

As per claims 10-11, these claims represent an apparatus performing a method corresponding to the method of claims 3 and 4, hence are rejected using the same arguments as

cited above in the respective claims (Fig. 3 – Note: O.S. device driver recognition cooperating with shell or windows API for signaling an hardware insertion is implicitly disclosed; col. 6, line 20 to col. 8, line 32; col 7, lines 30-50).

As per claim 12, Shih discloses communications with bus and multi-machine environment (Fig. 1); and Garney teaches that the expansion card comprises a transmitter/receiver unit and power amplifier (e.g. *device driver* - col.1, li.60 to col.2, li.4). At the time of the invention, it was a well-known concept to one of ordinary skill in the art that a power amplifier is commonly used in the output stage of a signal-producing device to isolate output impedance. Additionally, it was also well-known in the art that a driver acts as transmitter/receiver unit to control components of a specific computer resource, and that card like modem or game adapter card come with a speaker being amplified by a power amplifier. Hence if Garney (in combination with Shih) does not already provide a high frequency power amplifier at the output stage of the transmitting unit, it would have been obvious to a person of ordinary skill in the art to modify Garney's expansion card so that it does come with one such amplifier to generate audio or signal with frequencies capable of being amplified for securing distance transmission of signal or impedance matching purposes.

As per claim 13, Shih further teaches an apparatus for performing the method of claim 1 wherein the electronic device is a data processor (e.g. col. 6, lines 5-32).

As per claim 14, Shih further teaches a storing means for performing the method of claim 1 (e.g. *memory* - col.2, li.2; Fig. 12); all of whose limitations having been addressed above in claim 1.

As per claim 15, Shih discloses a method for loading the application software of an expansion card in an electronic device, said method comprising loading, starting and executing program modules in the device;

which expansion card can be coupled in a releasable manner to the device (Fig. 1; 3);
executing the loading of the expansion card application software is done in 2 phases (Fig. 2, 3);

wherein the first phase includes the loading and start-up of the basic module (e.g. *event monitor*, step 300 - Fig. 2; col. 6, lines 25-37); and

the second phase includes conducting the loading and start-up of the software application module (e.g. steps 315-320 – Fig. 3) when the expansion card is coupled to the electronic device (Fig. 3) and the basic module has already been loaded (Note: event monitor to see if insertion or removal event occurs reads on its being loaded prior to such event); the basic module and the card application being stored within the electronic device (Note: basic module and card application being loaded are construed as being stored in the loading process).

From claim 1, Shih discloses that the expansion software application to be loaded can be a user interface software or application modules (e.g. col. 6, lines 5-32) used in such handheld devices.

Shih does not explicitly disclose optionally stopping between the first phase and the second phase, but this is implicitly disclosed because (see Fig. 3) any time the user chooses to stop the event monitor at stage 300, the user can effect an manual interruption and enable the stop between the 2 phases, i.e. alternative by user to stop the installation before it starts reads on optionally stopping.

Nor does Shih specify that said user interface software is divided in a basic module and a user interface module, said basic module and said user interface module being separate parts of the same user interface software. But these limitations have been addressed in claim 1 above.

As per claims 16-20, refer to respective rejections of claims 2-6.

As per claim 21, this corresponds to claim 7, hence is rejected using the corresponding rejections as set forth therein; and further recites means capable of stopping the loading between the loading of the basic module and user interface module. This limitation has been addressed in claim 15 above.

As per claims 22-27, refer to respective rejections of claims 8-13.

As per claim 28, this is a storing means version of claim 15; hence is rejected using the corresponding rejections as set forth therein, the storing means being inherent in a processing unit like the computing system as taught by Shih.

Response to Arguments

7. Applicant's arguments filed 6/7/2006 have been fully considered but they are not persuasive. Following are the Examiner's observation in regard thereto.

(A) Applicants have submitted that Shih, 'the *autorun* program' is activated only after the insertion of the expansion card is detected, so that the autorun and the application are activated after the card step is inserted, thus unlike what is claimed wherein the basic module is loaded before the card is coupled (Appl. Rmrks, pg. 11, bottom, pg. 12, top). The rejection set forth in USC §112, 2nd paragraph has pointed out how unclear the basic module loading is. This loading of the first phase does not include any teaching 1) concerning the nature of the loading nor does it clearly states that it does not operate via the expansion card; 2) that it necessarily enforces that

the basic module is actually loaded from within the memory of device by the time the first phase starts, as opposed to the loading coming from the expansion card and its resident UI software. The claim recites what appears to be one loading process and even though it is described in 2 phases, it is indefinite as to the state of the UI software components being stored when the loading starts (*), as set forth in the 112, 2nd rejection. This very same loading step for enforcing that the UI module can only be loaded (second phase) from the card only after the basic module has been loaded and executing, does not justify to the construction perceived in (*) whereby both the UI module and the basic module are being (currently?) stored in the device; making what appears to be the Applicant's conviction that the basic module is operating from the electronic device and not coming from the same card being inserted at the start of the second phase a alleged feature that is not clearly seen from the claim as it is recited. The claim does not enforce that only the basic module is loaded from within the electronic device and that the loading step is actually loading just the UI module from the card, which proves not to be the case from (*). There is nothing in the claim to enforce that the autorun by Shih actually reads on the basic module as alleged by Applicant; and that Shih's Event monitor cannot read on basic module, in view of the rationale using Garney's teaching to enable an obvious rationale that would make the event monitor by Shih a separate part of the software application provided in the expansion card. The claim amounts to a basic module operating first in the device as it is loaded for detecting the insertion card, whereupon it will enable the loading from the card the desired software application. Shih's event monitor reads on the snooping of the loaded basic module; and this enable the detection and activation of the main software from the inserted card as required from the claimed second phase. The basic module as claimed does not preclude Shih's event monitor

program from reading into it; nor does the role of the basic module as claimed necessarily be that of Shih's autorun as argued. If the role of the basic module is such that the basic module is part of the UI application (target software to load) so that it effectuate both the card insertion detection and loading of said software onto the device, then Shih's process can be enhanced to include the teachings by Garney whereby the expansion software in the card has a first module being loaded to operate the steps recited in the second phase. The 103(a) rejection is purported to address such rationale, given the motivation and reasons to combine. To rebut Shih's, the Applicant is burdened to address the rejection as a whole not a piecemeal analysis of reference taken separately.

(B) Applicant has submitted that Garney's stub is read from the card and that when activated the full device driver is not transferred onto the device but merely executed from where it is stored, in the card; and the stub is executed after the card insertion (Appl Rmrks, pg. 12, 2nd para). The rejection has shown why by providing a module distinct (say a stub module) from the main module (say, a software driver) to install, the endeavor as to accommodate the specifics of target computer system or different architecture machine diversity with the application software to be loaded by Shih can be enhanced by Garney, who supplies a separate module first for observing whether the target system has what it takes to be compatible to be using the application. In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). Further, the limitation recited as 'the basic module loading the user interface module' is not strongly conveying that first there is

transfer from the card into the device storage; and then loading from said storage into execution memory of the device for execution. The stub by Garney enables the driver to be executed reads on the loading for execution; and such is sufficient to combine with the event monitor by Shih to effect the required loading as recited in the second phase of the claim, absent any specifics about how this loading is being implemented. The Applicant's argument about the driver being executed while resident on the card is therefore not persuasive. The stub by Garney is purported to address the limitation that the UI software has 2 parts, one of which being activated first to detect the insertion of the card and enable the loading of the UI module. Shih is not exactly providing a same software product with 2 parts cooperating to enable the loading of the main part after the first part has been activated. Garney provides one same software product having 2 parts (which Shih does not show), one of which being activated first to enable the loading of the second part phase. Besides, the claim does not make it clear that the loading of the basic module has to be done absolutely without the card being inserted at all. The newly added limitation recited as 'and being stored within the electronic device' is not giving any definite clue about where the basic module is located when the loading starts in the first phase -- when both modules, this module and the UI module, are recited as being stored together (see USC 112, 2nd para); nor does the claimed 'loading and start-up of the basic module' in the 1st phase necessarily precludes any card insertion at the time of such loading(e.g. such that the card can be re-inserted later at a time where the basic module has already been running for the second phase to be enabled). The rejection has set forth the endeavor and benefits of such combination of analogous methodologies by both references. In regard to applicant's arguments against the references

individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references.

(C) Applicant has submitted that Shih's event monitor is not the claimed basic module when Applicant's basic module is 'a basic module of a particular expansion card', not from a the expansion card'; and that Shih's generic event monitor is not this basic module being already loaded when the card is inserted (Appl. Rmrks, pg. 13, first 2 para). The claim is not clear about the location of the basic module when the first phase starts; in view of the interpretation that both the software modules are being stored in the electronic device when the loading originally started; and that the loading of the UI module is effectuated only via card insertion. Whether the claim enforces that only the basic module resides on the device when the loading process gets started or that the expansion card only stores the UI module when the second loading phase starts, based from the way the claim has been recited, that is quite indefinite. The Applicant's argument is based on features that appear not to come from the subject matter being interpreted from the claim: the claim does not enable the understanding that (i) the basic module even though is from the same UI software but has already been installed in the electronic device and is by no means present in the expansion card; and that the UI module being resident on the expansion card is activated by this basic module only by means of the insertion of the second phase. From the analysis as observed from the USC 112, 2nd para, both the UI module and the basic modules are part of the expansion card UI software, and the loading can be interpreted as being done in the following alternative: the card is possibly inserted first to activate the basic module; and upon subsequent re-insertion of the card, the UI module would be loaded – second phase -- for execution using the detection and action from the basic module, activated in the first

phase. The claim does not preclude that there can be more than one card insertions for the UI module to be loaded; nor the claim is definite as to whether the loading requires transfer of data first then execution after when data has been stored in the electronic device. Arguing that the basis module is not a universal generic program would not solve the deficiencies of the claim as exhibited in the USC 112, 2nd paragraph; because the basic module as claimed does not preclude a piece of software running in Shih's environment to activate the loading as intended by the second phase, that is, the claim does not make it clear how the activation of the basic module is done to enforce otherwise. Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

(D) The argument about Garney's stub (Appl. Rmrks pg. 13, bottom, pg. 14, top) being inserted after the card is inserted is to be referred to section B above. Likewise, the argument about how the loading of the UI module by the basic module is not taught has to be referred thereto also.

(E) Applicant has submitted that the references rely on software modules being stored on the card itself and Shih's event monitor executes from the shell while the autorun remains on the card, contradicting with event monitor and autorun being stored on that card (Appl. Rmrks, pg. 14, middle to pg. 15, top). This issue about autorun and event monitor has been addressed in section A above; that is, the autorun cannot be analogized with the basic module as claimed or as proffered in the argument.

Art Unit: 2193

(F) Applicant has submitted that the Examiner has not shown ‘loading’ and ‘start-up’ (pg. 15, 2nd para) and that both Shih’s autorun and installation software reside in the card; not that both the basic module and the UI module are stored within the device as claimed (pg. 15, 3rd para) The rejection has addressed the loading and start up as recited; and the USC 112, 2nd para has helped set forth the indefiniteness of the language of the claim thereby allowing the rejection to be effectuated, i.e. based on the interpretation as set forth above in the USC 112 rejection: see section A.

(G) As for the argument about Garney’s driver code being executed from the card itself (pg. 15, bottom) this is to be referred to section B. The argument that both the basic module and the UI module are stored within the device has been addressed in section A, B in light of the USC 112, 2nd para.

(H) As per the argument raising Examiner’s failure to show the stopping limitation of claim 15, it is interpreted from the ‘optionally stopping’ an event based on uncertainty or unpredefined likelihood; and the note as set forth in the rejection has addressed such uncertainty posed by the way this ‘stopping’ limitation is claimed. A limitation has to set forth solid metes and bounds, and this ‘optionally’ characteristic does not lend to a requirement of any weight that would necessitate a very sturdy evidence to anticipate; i.e. the rejection using the user option to stop the process at will anytime will stand. Against the observation that there is no motivation to combine (Appl. Rmrks, pg. 17), the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art.

See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, the rejection has set forth the analogous endeavor of the references, the actions and underlying purposes and based on which, using knowledge of one of ordinary skill in the art when faced with such teachings or underlying advantages therefrom, has provided the combination in view of such common purpose in order to solve a problem, in view of the benefits that would be derived from such knowledge (refer back to rejection). Burden is on the Applicant to particularly point from the cited portions where the deficiency (e.g. adverse effects of joined teachings) to the 103(a) rationale occurs; and why so.

(I) Applicant has submitted that Shih does not fulfill the limitation of claim 2 (Appl. Rmrks, pg. 18, middle). The rejection has analogized claim 2 with the action of the event monitor to enable the detection and subsequent installation of the main software, thus the sequence constitute a basic module controlling the execution of the second phase. Applicant is not consistent in his argument about Shih's event monitor. At one time, it has been argued that the event monitor cannot read on the control action performed by the claimed basic module; at others, it has been argued that Shih's *autorun* for staying in the card cannot read on the role played by the basic module. Detecting an event contributes to what is called control of the flow of events or actions that follow; and Shih's event monitor has contributed in just that part of control; and the argument (pg. 18, bottom) against claim 3 in regard to 'initiated from the basic module' is seen as being mapped with the event monitor role in its triggering role for the subsequent flow of actions in Shih's process.

(J) As for the arguments against claim 5 (Appl. Rmrks, page 12), the rejection will stand; because the argument sound like mere allegation to the contrary ('is not the same as') of what

has been cited in the rejection without explanation as to why. As for argument against claim 12 (pg. 19, middle) in which it is argued that no reference teaches a transmitter/receiver as claimed. It is noted that there is not specifics in the claim to enforce a distinct implementation of transmitter or receiver that would clearly distinguish from the modem or PCMCIA adapter as set forth in the rejection, notwithstanding the understanding that network communication frequencies are not low frequencies. Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

Thus, the rejection will stand rejected as set forth.

Conclusion

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571)272-3719.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Art Unit: 2193

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Tuan A Vu
Patent Examiner,
Art Unit 2193
July 03, 2006